Clickbait intensity prediction and its relevance to Fake News

Information Retrieval and Extraction Project

Team 22 **Mentor: Vijaya Saradhi**

Archit Jain	Aryan Jain	Palash Sharma	Pulkit Gupta
(2019101053)	(2019101056)	(2019101082)	(2019101078)

Links

i. GitHub Repo: https://github.com/Architjain128/Clickbait-FakeNews
ii. Dataset 1: https://zenodo.org/record/5530410#.YXrgehzhU2w
iii. Dataset 2: https://github.com/laxmimerit/fake-real-news-dataset/tree/main/data
iv. Presentation Video: https://www.youtube.com/watch?v=yRhEFS0AUfU

Related Work

This is the first paper to model clickbait identification as a regression problem considering predicting clickbait intensity as a real number between 0 and 1 (all previous papers consider it a classification problem considering predicting whether it is clickbait or not). In this paper, the authors have used transformers to convert text to vectors before applying it to several regression models.

Fake news classification is a famous problem in the NLP arena and mostly involves using the traditional classification algorithms on word embeddings. Advanced papers in this area of research involve fine-tuning Language Models among other niche techniques. For our scope of the project, we have omitted fine-tuning any pretrained model (due to lack of resources).

This paper presents a system for the detection of the stance of headlines about the corresponding article bodies. The approach can be applied in fake news, especially clickbait detection scenarios. An n-gram matching of the lemmatised input (headline or article) is applied first to decide whether a particular headline/article combination is related or unrelated. Further classification is done using Mallet's Logistic Regression classifier to classify the related pairs into "agree" or "disagree".

This paper aims at finding a relation between click baits and fake news. The approach followed involves training a binary classifier for click baits, labelling sources of news article in terms of clickbait intensity percentage and then using this as a feature to train a binary fake news classifier. This method assumes that a clickbait intensity score can be generalized for a media house which is not true.

Baseline Models

As a strong baseline:

a. We used Linear Regression using spaCy vectorizer as our baseline for clickbait intensity prediction because linear regression is the most basic regression algorithm, also spaCy is easy to implement and fast.

b. We used Naive Bayes Classifier using spaCy vectorizer as our baseline for fake news detection (which was a binary classification problem) as it is easy to build and extremely fast as compared to another classifier.

Dataset

- **Clickbait Intensity**
  We have used Webis Clickbait Corpus 2017 dataset for predicting clickbait intensity. The dataset includes fields like "postTimestamp", "postText", "postMedia", "targetTitle", targetKeywords", "targetDescription", "targetParagraphs", "targetCaptions" etc and we have used "postText" as a feature to train the regression model and "truthMean" as the value for clickbait intensity.
- **Fake News Detection**
  We have used the dataset named fake-real-news-dataset which has two separate files, one containing all the articles which are labelled as fake and other one labelled with true, each containing an article title, article text description, its subject along with timestamp of article but we have used title, text and label of article (Fake/Real) along with the clickbait intensity score predicted from model formed in Clickbait Score model. The dataset has equal(almost) number of fake and non-fake data, which would ensure a better model fitting than the other case.

Architecture

Our solution intends on classifying a news article as fake or real considering the clickbait intensity of the title of the article. This approach can be divided into two parts, training a regression model to predict clickbait intensity, and then using this clickbait intensity as one of the features to train a classification model for fake news prediction.

Clickbait Intensity Prediction: -

- Webis dataset is first pre-processed to isolate the clickbait intensity and postText(title). Then the pre-processed title in this dataset is then passed to a vectorizer to convert titles into vectors. We have worked with Spacy Vectorizer [i] as a word embedding pretrained model and RoBERTa as a transformer model. Note that both these models give a vector for a title, and we used only one of them at a time.
- The vectorized data is then fed to a regression model to train. The parameters of the model are fine-tuned to minimize the MSE for the model. We have split the dataset to use 80% of the dataset for training the model and 20% to test it.

Considering this prediction as regression problem we have used a bunch of regression models.

Regression models that we have used are-

- **Linear Regression:** Independent variables are related linearly, and least square sum is used for model fitting.

- **Ridge Regression:** similar to linear regression, where penalty term is added with least sum of squares to gear the correlations between the independent variables.
- **Random Forest Regression:** a supervised learning algorithm that uses ensemble learning method for regression.
- **Gradient Boosting Regression:** relies on the intuition that the best possible next model, when combined with previous models, minimizes the overall prediction error.
- **Adaboost Regression:** a meta-estimator that begins by fitting a regressor on the original dataset and then fits additional copies of the regressor on the same dataset but where the weights of instances are adjusted according to the error of the current prediction.
- **Extreme Gradient Boosting Regression:** efficient open-source implementation of the gradient boosted trees algorithm.
- **Stacking Regression Model:** that includes gradient boosting, ridge regression, random forest, adaboost, extreme gradient boosting as estimators and linear regression as final estimator.

Fake News Prediction: -

- The fake-real news dataset is pre-processed in the same way as clickbait intensity and the article text and title are passed to a vectorizer (same as above) to obtain a vector for each article respectively.
- Additionally, the title of the article is passed to the regression model (trained before) to obtain a clickbait intensity score. This score along with the previously obtained vectors together forms a set of 2k + 1 features (where k is the dimension of a single vector).
- The set of features obtained from the previous step are used to train the classification model for binary classification of authenticity of a news article. The model is then fine-tuned by altering the parameters to maximize the F1 score.

Considering this prediction as classification problem we have used a bunch of classification models.

Classification models we have used are-

a. **Logistic Regression:** is a predictive analysis algorithm based on the concept of probability.
b. **Adaboost Classifier:** combine multiple “weak classifiers” into a single “strong classifier”.
c. **Naive bayes:** is a supervised learning algorithm based on bayes theorem.
d. **Decision Tree Classifier:** is a tree structured classifier starting from root node and split the data on feature that result in reduction in uncertainty.
e. **Random Forest Classifier:** is a classification model consisting of multiple decision trees.
f. **Linear Support Vector Classifier:** chooses extreme points/vectors that help in creating best decision boundary.
g. **Passive Aggressive:** is an online learning algorithm where a model is trained and deployed in production in a way that continues to learn as new data sets arrive.
h. **K-Nearest Neighbor:** assumes that similar things exist in close proximity i.e., similar things are near to each other.
i. **Extreme Gradient Boosting Classifier:** belongs to the family of boosting algorithm using gradient boosting framework as its core.
j. **Stacking Classifier Model:** that includes decision tree, extreme gradient boosting, random forest, linear support vector classifier, adaboost as estimators and logistic regression as final estimable

***Stacking** is an ensemble machine learning algorithm which stacks the output of individual estimator and use a regressor to compute the final prediction allowing us to use the strength of each individual estimator by using their output as input of a final estimator.*

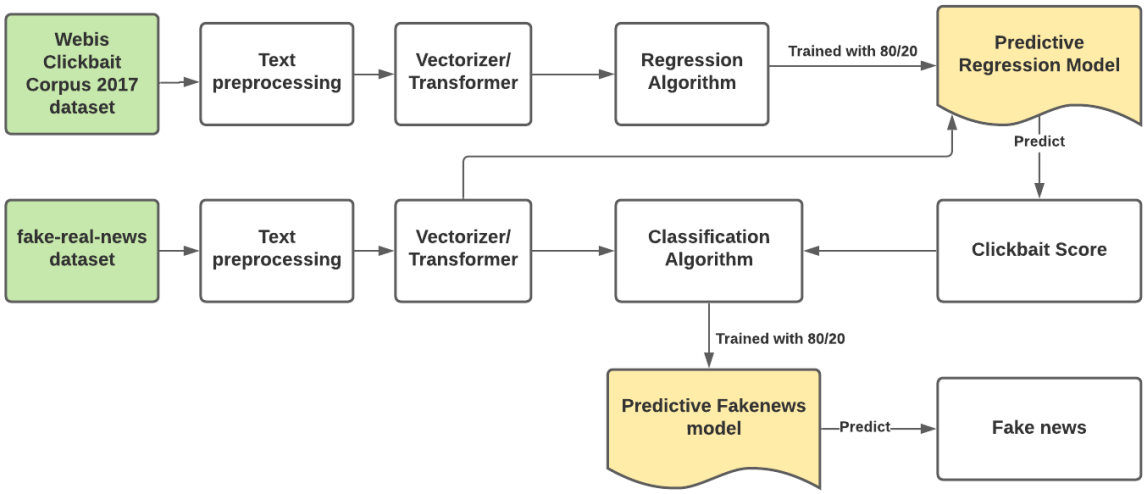

Figure 1: Workflow of out Architecture

Prepossessing

- **Cleaning Text**

  This includes converting text to lowercase, removing non-alphanumeric characters, and using lemmatization and removing ambiguous or incomplete data.

- **Vectorization**

  Vectorization or word embedding is the process of converting text data to numerical vectors. Later those vectors are used to build various machine learning models. Vectorization is a sophisticated way of including the information contained in the text to machine learning models.

  Vectorizers we have used:

1. **spaCy**: A pre-trained word embedding model is loaded, using which the required text is vectorized. The spaCy vectorizer takes a sentence as an input and converts it into vector representation. We used 2 types of pre-trained model. One was small (with just 96 dimensions) and other was large (with 300 dimensions). The model trained on larger dataset produced best results for our case.
2. **Google's Word2Vec**: Instead of sticking to just one vectorizer, we thought of trying with other vectorizer also namely Google's Word2Vec embedding model. This model takes one word at a time and returns a particular vector for that word unlike spaCy where we get vector for complete sentence. We used tf-idf score averaging to obtain the vector for the sentence. The vectors obtained from this approach was also of 300 dimensions.

Transformers

RoBERTa Transformers it stands for Robustly optimized BERT approach. RoBERTa, is a retraining of BERT with improved training methodology, 1000% more data and compute power. RoBERTa removes the Next Sentence Prediction (NSP) task from BERT's pre-training and introduces dynamic masking so that the masked token changes during the training epochs. Importantly, RoBERTa uses 160 GB of text for pre-training, including 16GB of Books Corpus and English Wikipedia used in BERT. The additional data included Common Crawl News dataset (63 million articles, 76 GB), Web text corpus (38 GB) and Stories from Common Crawl (31 GB). The dimension of the vectorized output was 1024, and it also takes contextual meaning of the sentence into account, so it's better than trivial vectorizers.

Evaluation metrics

- For clickbait intensity prediction, model predict intensity as a real number between 0 and 1 so we have used Mean Squared Error (MSE) for clickbait intensity prediction, which was a regression problem. A lower MSE for a model signifies a better fit.
- As our fake news detection was a binary classification problem, so we used F1 score as metric here as it is a measure of test's accuracy. F1 score gives a realistic reflection of the performance of a classification model. A higher F1 score is considered better.

Results

Table 1: Mean squared error for different algorithm used for regression model

Vectorization Method	Method	MSE
spaCy	Ridge Regression	0.04171198
	Linear Regression	0.04166450
	Adaboost Regression	0.15428521
	Random Forest Regression	0.15960722
	Gradient Boosting Regression	0.03749024
	Extreme Gradient Boosting Regression	0.03681164
	Stacking Regression Model	**0.03390061**
RoBERTa	Ridge Regression	0.03072690
	Linear Regression	0.03133114
	Adaboost Regression	0.19042188
	Random Forest Regression	0.15551844
	Gradient Boosting Regression	0.03469860
	Extreme Gradient Boosting Regression	0.03335698
	Stacking Regression Model	**0.03047084**

Table 2: F1 score and Accuracy for different algorithm used for classification model

Method	Without Clickbait score		With Clickbait Score		Status
	F1 Score	Accuracy	F1 Score	Accuracy	
Logistic Regression	**0.9784**	**0.9793**	**0.9974**	**0.9970**	F1 Score Improved
Adaboost Classifier	0.9332	0.9358	0.9384	0.9405	F1 Score Improved
Naive Bayes Classifier	0.8656	0.8656	0.8652	0.8586	No Improvement
Decision Tree Classifier	0.8617	0.8700	0.8697	0.8770	F1 Score Improved
Random Forest Classifier	0.9318	0.9340	0.9333	0.9366	F1 Score Improved
Linear Support Vector Classifier	0.9770	0.9755	0.9770	0.9783	Accuracy Improved
Stacking Classifier Model	0.9511	0.9539	0.9505	0.9534	No Improvement
Passive Aggressive Classifier	0.9724	0.9730	0.9724	0.9731	Accuracy Improved
K-Nearest Neighbor Classifier	0.9129	0.9147	0.9133	0.9151	F1 Score Improved
Extreme Gradient Boosting Classifier	0.9544	0.9533	0.9582	0.9523	F1 Score Improved

- For Clickbait intensity prediction, Stacking Regression model (that includes gradient boosting, ridge regression, random forest, adaboost, extreme gradient boosting as estimators and linear regression as final estimator.) works best with minimum MSE of 0.03047084 using RoBERTa as a vectorizer.
- For fake news classification, Logistic Regression Classifier outperforms all other classifiers in both with and without clickbait with F1-score as 0.9974 and 0.9784 respectively.

Analysis

For Clickbait Prediction

- Comparing the vectorizers, spaCy being just 300 dimensions was significantly faster than RoBERTa-large which has 1024 dimensions. Since RoBERTa has more dimensions, it is able to capture more features in text. RoBERTa, like BERT, uses bidirectional positions to help the algorithm understand the meaning of ambiguous language in text by using surrounding text to establish context enabling a better understanding of the text. Hence, RoBERTa proved more useful than spaCy.

- Adaboost is simply a linear combination of regressors which experience overfitting in high dimensional spaces. Also, the same is seen in our evaluation of the classifier. SpaCy vectors had 300 dimensions and the MSE reported for that is ~0.154. But for

RoBERTa vectors which has 1024 dimensions, the MSE comes out to be ~0.190. So we saw overfitting in this case.

For Fake news Prediction

- Decision Tree Classifier does not perform well (in comparison to other classifiers) for fake news prediction because firstly, it's a greedy model, which implies that it will make the most optimal decision at each step but does not consider the global optimum. However, choosing the best result at a given step does not ensure we will head down the route that will lead to the optimal decision when we make it to the final node of the tree, called the leaf node.

  Secondly, it is highly prone to overfitting with its depth. This is due to the amount of specificity we look at leading to smaller sample of events that meet the previous assumptions. This small sample could lead to unsound conclusions.

- Logistic Regression performs better than naive bayes, as naive bayes expects all features to be independent but in case of fake news prediction all features are related to each other and hence the result.

- Using clickbait score as one of the features for fake news detection, F1 score increases which signifies the relevance of clickbait score to fake news. This is expected because to make an article attractive, clickbait titles are common, and they tend to be fake more often.